Engenharia de Controle e Automação

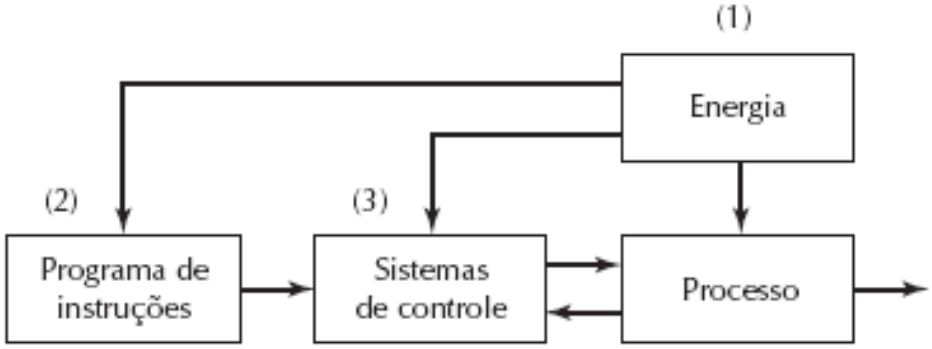
Informática Industrial I

Introdução ao Sistema de Automação e a Linguagem 61131

1o Semestre 2020

Automação e Tecnologia de Controle

Sistema de Controle :



- Executa as instruções de forma a realizar alguma operação de produção

Programa de Instruções

Imagine um cenário onde você está no controle industrial

- trabalhando com diversos controladores de marcas diferentes que possuem diferentes linguagens de programação
- E você tem diferentes nível de engenheiros e operadores de manutenção no chão de fábrica
- O que esta errado?

Fora da Selva...

- A atual variedade de problemas pode ser amplamente reduzida através da padronização

- O padrão IEC 61131

"A melhor coisa que aconteceu ao controle industrial" - Sugar Lantic on Automation Maillist

Padrão IEC 61131

- Para atender as demandas da comunidade industrial um grupo da International Electrotechnical Commission(IEC) (em Genegra) analisou e especificou o projeto de Controladores Programáveis definindo:
  - Hardware, instalação, teste, documentação, programação e comunicação.
- 1ª especificação - IEC 1131 - de 1992
- Atual - IEC 61131

Estado atual da Norma IEC 61131

Parte	Titulo	Conteúdo	Publicação
IEC 61131-1	General information	Definições iniciais do conceito	V2.0 (2003)
IEC 61131-2	Equipment requirements and tests	Testes de verificação e fabricação	V3.0 (2007)
IEC 61131-3	Programming Languages	Estrutura de Software do CP, linguagens e execução de programa	V3.0 (2013)
IEC 61131-4	User Guidelines	Orientação para seleção, instalação e manutenção de CPs.	V2.0 (2004)
IEC 61131-5	Messaging service specification	Funcionalidades de comunicação com outros dispositivos.	V1.0 (2000)

http://www.plcopen.org/pages/tc1_standards/

Estado atual da Norma IEC 61131

Parte	Titulo	Conteúdo	Publicação
IEC 61131-6	Functional Safety	Especifica requerimentos para CPs, para uso em sistemas de segurança.	V1.0 (2012)
IEC 61131-7	Fuzzy control programming	Funcionalidades de software para utilização com lógica Fuzzy	V1.0 (2000)
IEC 61131-8	Guidelines for the application and implementation of programming languages	Orientação para implementação do 61131.	
IEC 61131-9	Single-drop digital communication interface for small sensors and actuators	Este padrão é conhecido como IO-Link e cobre interfaces de comunicacao.	Versão Draft.

IEC 61131-3 – Linguagens de Programação

Programação de controle industrial

Suporte para pessoas com diferentes níveis de conhecimento...

IEC 61131-3 Standard

Elementos Comuns

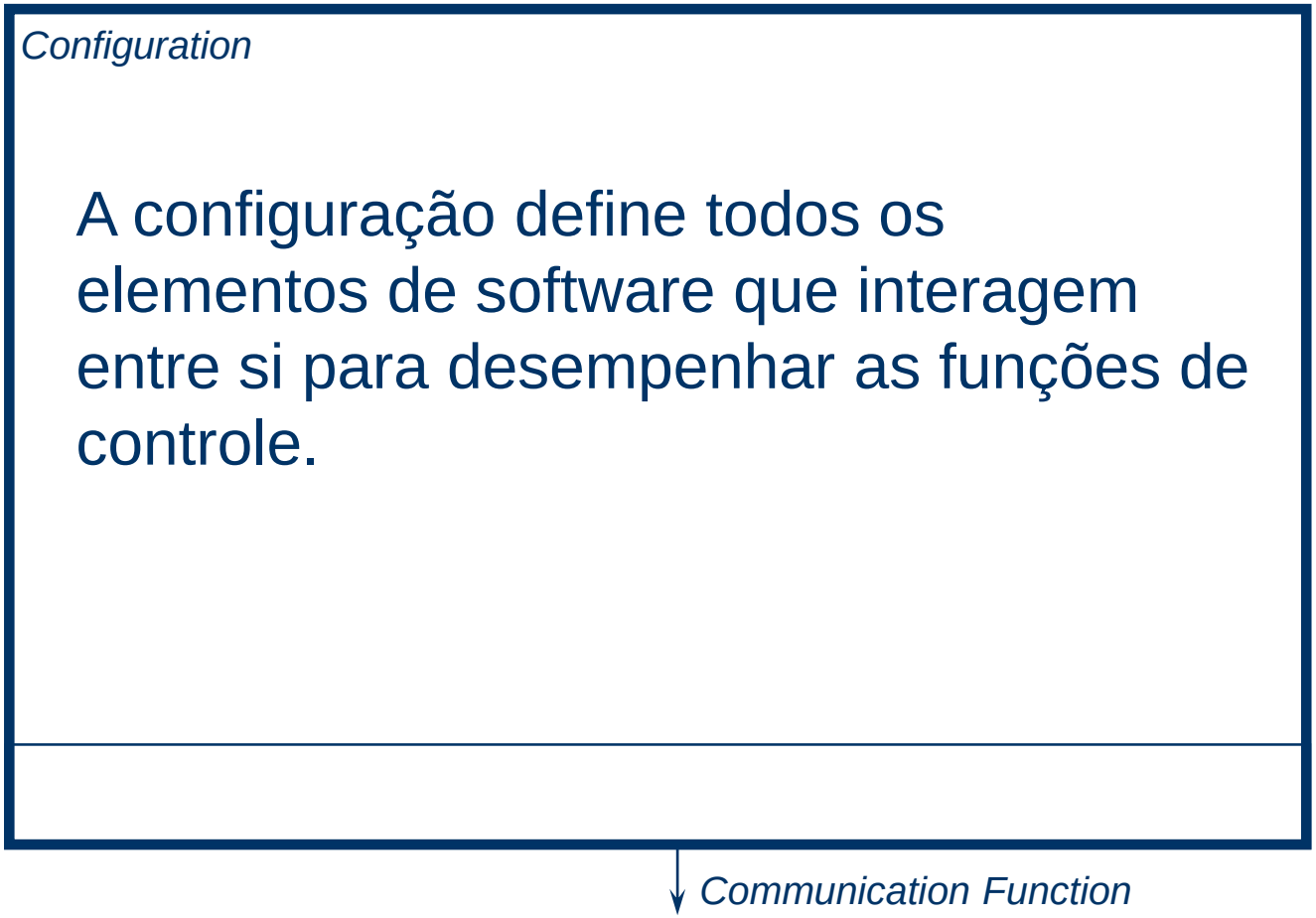
Linguagens de Programação

IEC 61131-3 : Elementos Comuns

Modelo de Software

- Configuração
- Recursos
- Tarefas
- Programas

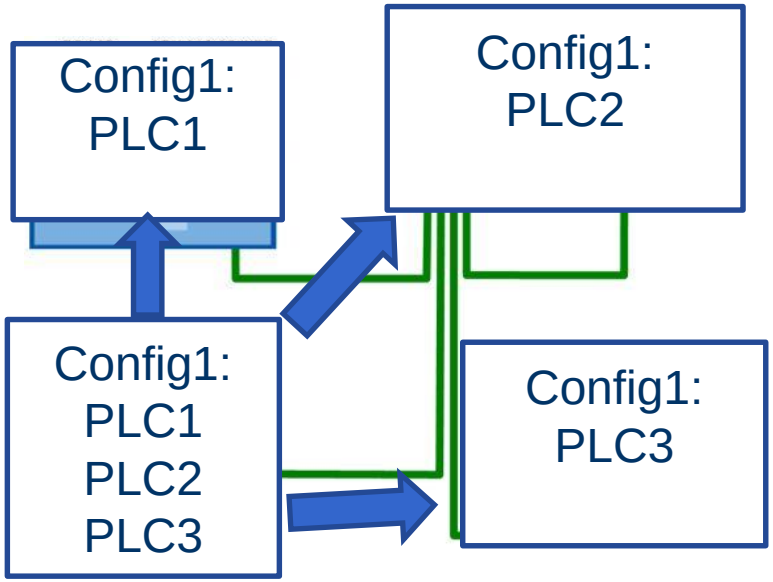
IEC 61131-3 : Elementos Comuns
Modelo de Software



IEC 61131-3 : Elementos Comuns
Modelo de Software

Configuração:

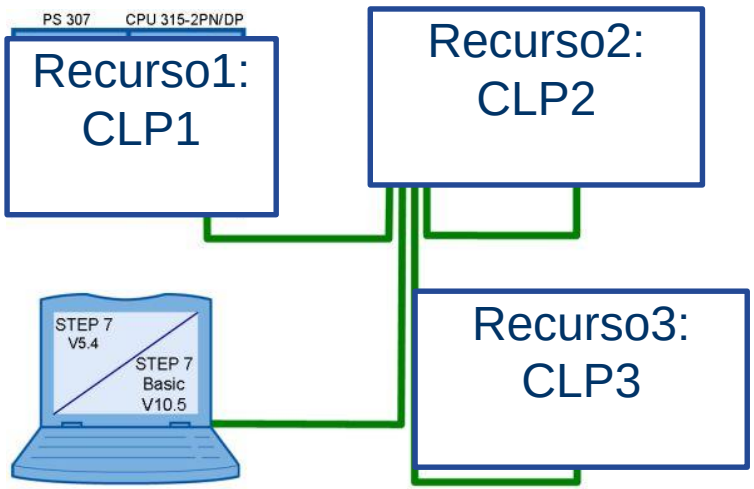
- Uma configuração pode corresponder a 1 ou mais CLPs.
- Para sistemas mais complexos pode existir vários CLPs de diferentes fabricantes.



IEC 61131-3 : Elementos Comuns
Modelo de Software

Recurso:

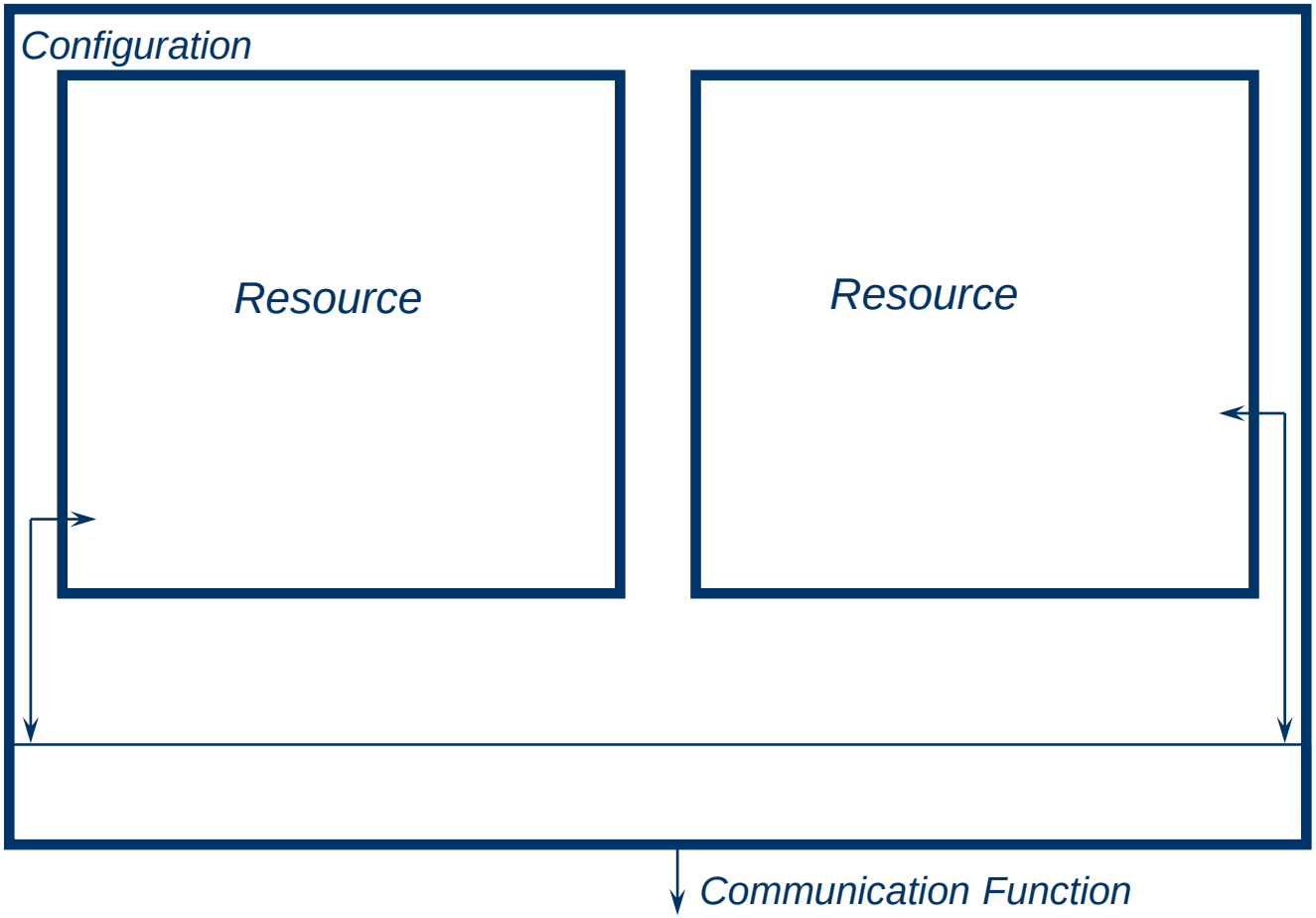
- Qualquer elemento com capacidade de processamento, responsável pela execução dos programas.
- Cada recurso pode ter um ou mais programas
- O recurso pode existir fisicamente ou ser apenas uma máquina virtual.



Exemplo – PC Based Solution Siemens

Exemplo de recursos distintos seria em um computador pode existir uma HMI e um SoftPLC (PC Based Solution).

IEC 61131-3 : Elementos Comuns
Modelo de Software

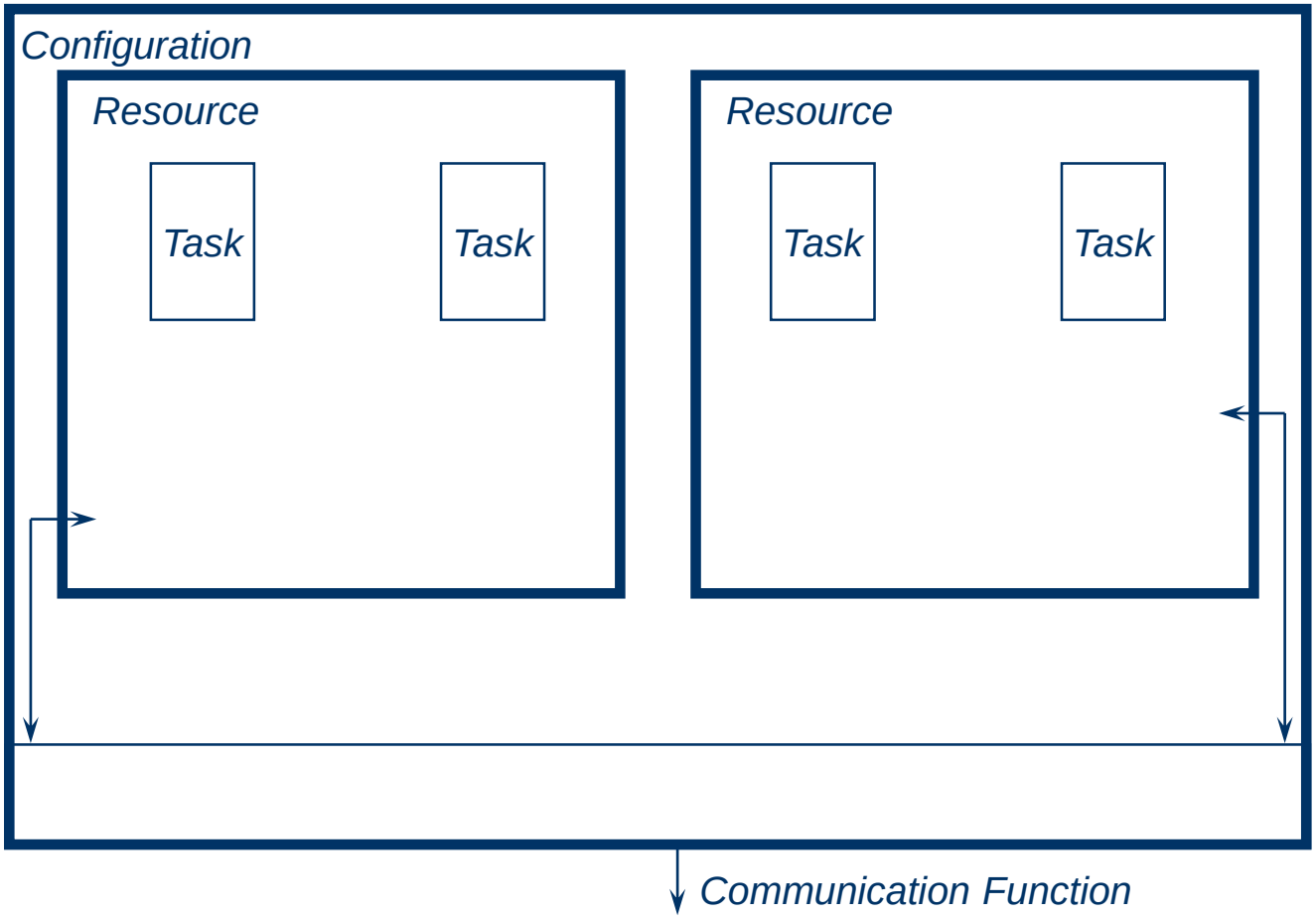


IEC 61131-3 : Elementos Comuns
Modelo de Software

Tarefas:

- Pode ser configurada para controlar a execução de programas ou blocos funcionais de forma periódica ou gatilhada com disparo de eventos (triggers).
- A norma não define os mecanismos de execução dos elementos de software mas define os comportamentos de partida e parada.

IEC 61131-3 : Elementos Comuns
Modelo de Software

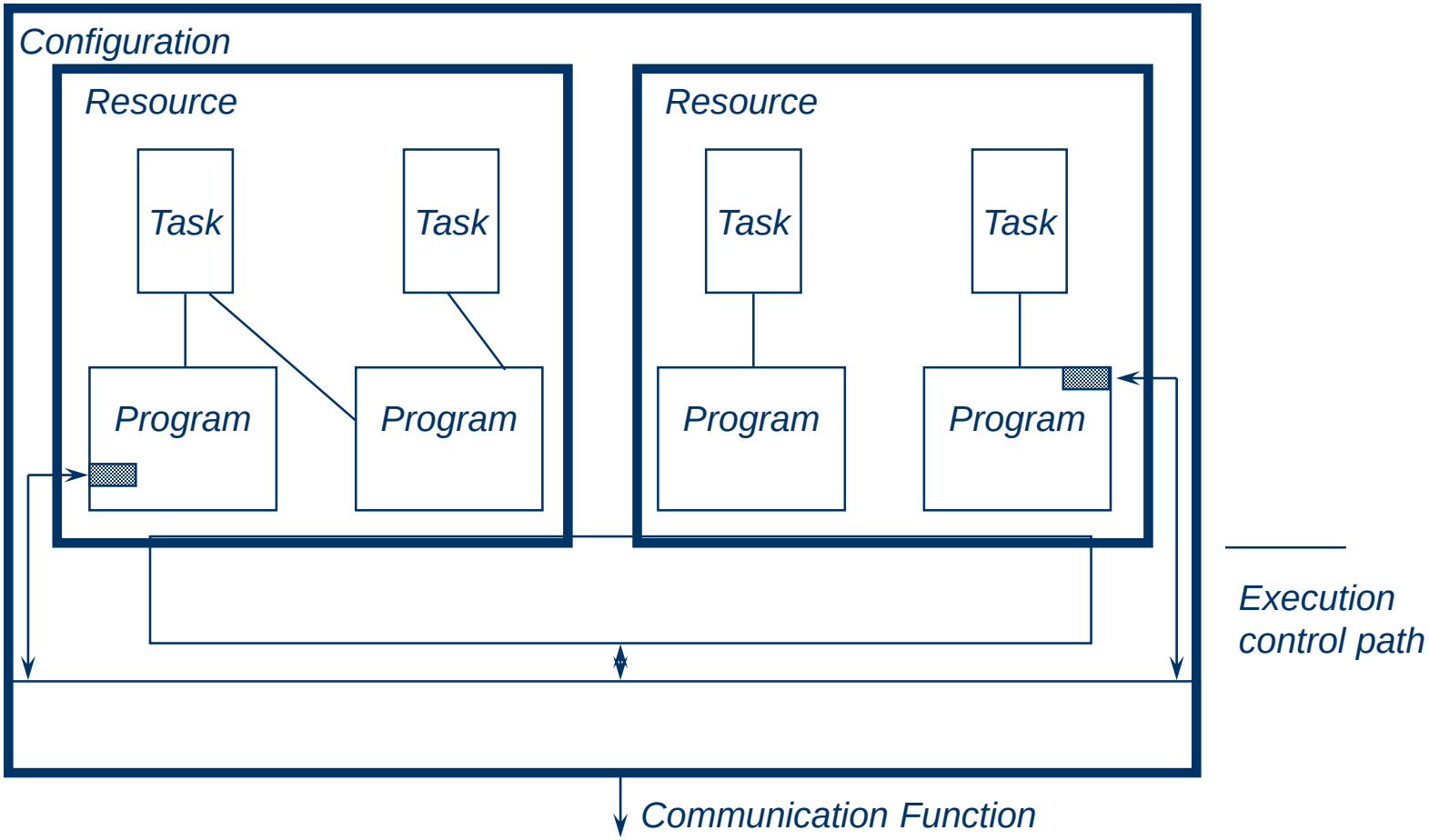


IEC 61131-3 : Elementos Comuns
Modelo de Software

Programa:

- Pode ser construído a partir de diferentes elementos de software, cada qual escrito em qualquer uma das linguagens 1131.
- Um programa pode acessar diretamente as variáveis de E/S e comunicar com outros programas.

IEC 61131-3 : Elementos Comuns
Modelo de Software

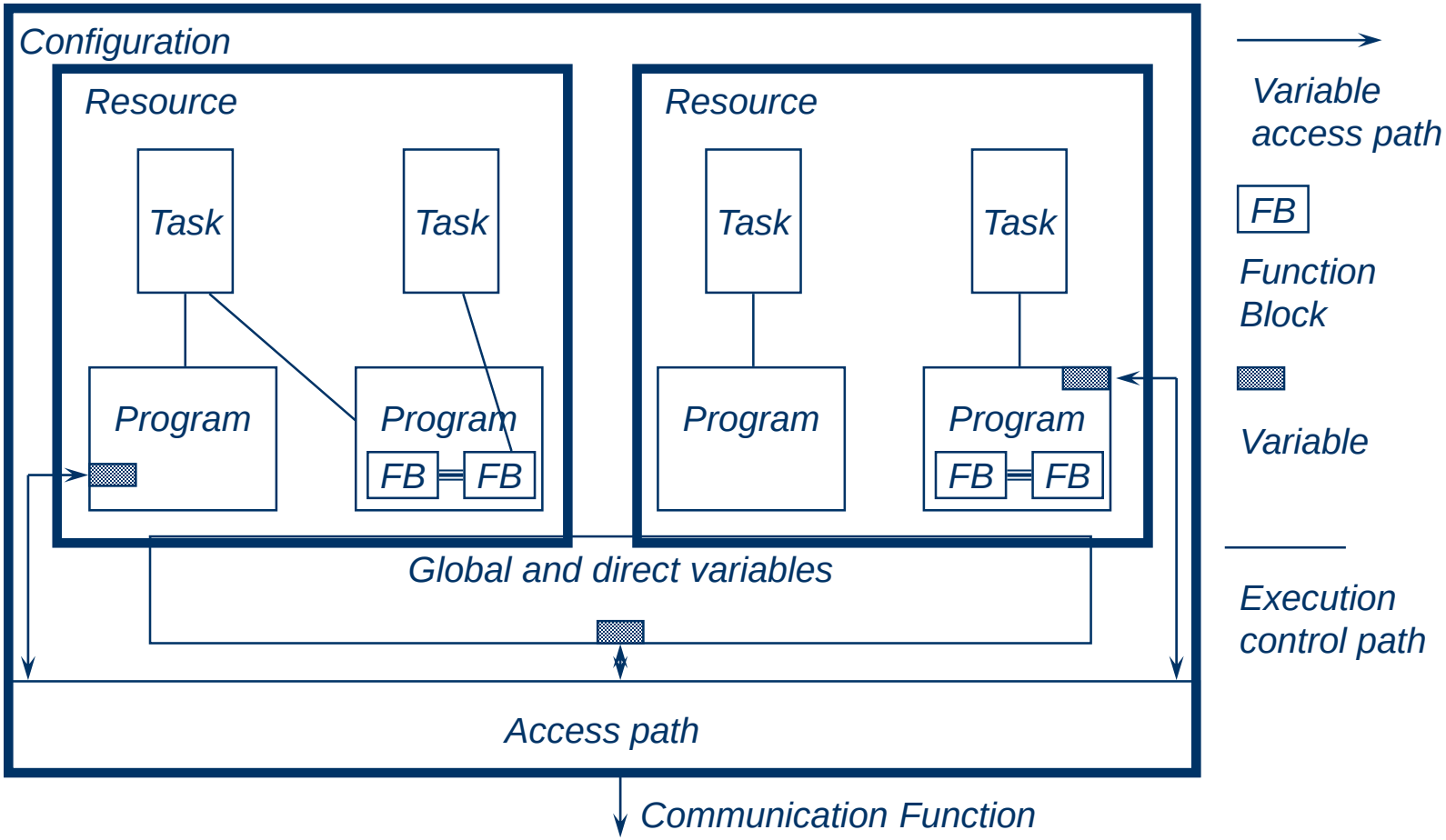


IEC 61131-3 : Elementos Comuns
Modelo de Software

Blocos Funcionais:

- Possuem um conjunto de dados que pode ser alterado por um algoritmo interno.
- Este conjunto de dados é mantido na memória para uma determinada instância do bloco funcional.
- O uso de blocos funcionais permite o projeto de software de forma hierárquica e estruturada.
- Podem ser utilizados para criação de elementos de software totalmente reutilizável, desde a utilização de outros blocos funcionais mais simples ate programas complexos.

IEC 61131-3 : Elementos Comuns
Modelo de Software



Modelo de Software IEC

Variáveis Locais ou Globais:

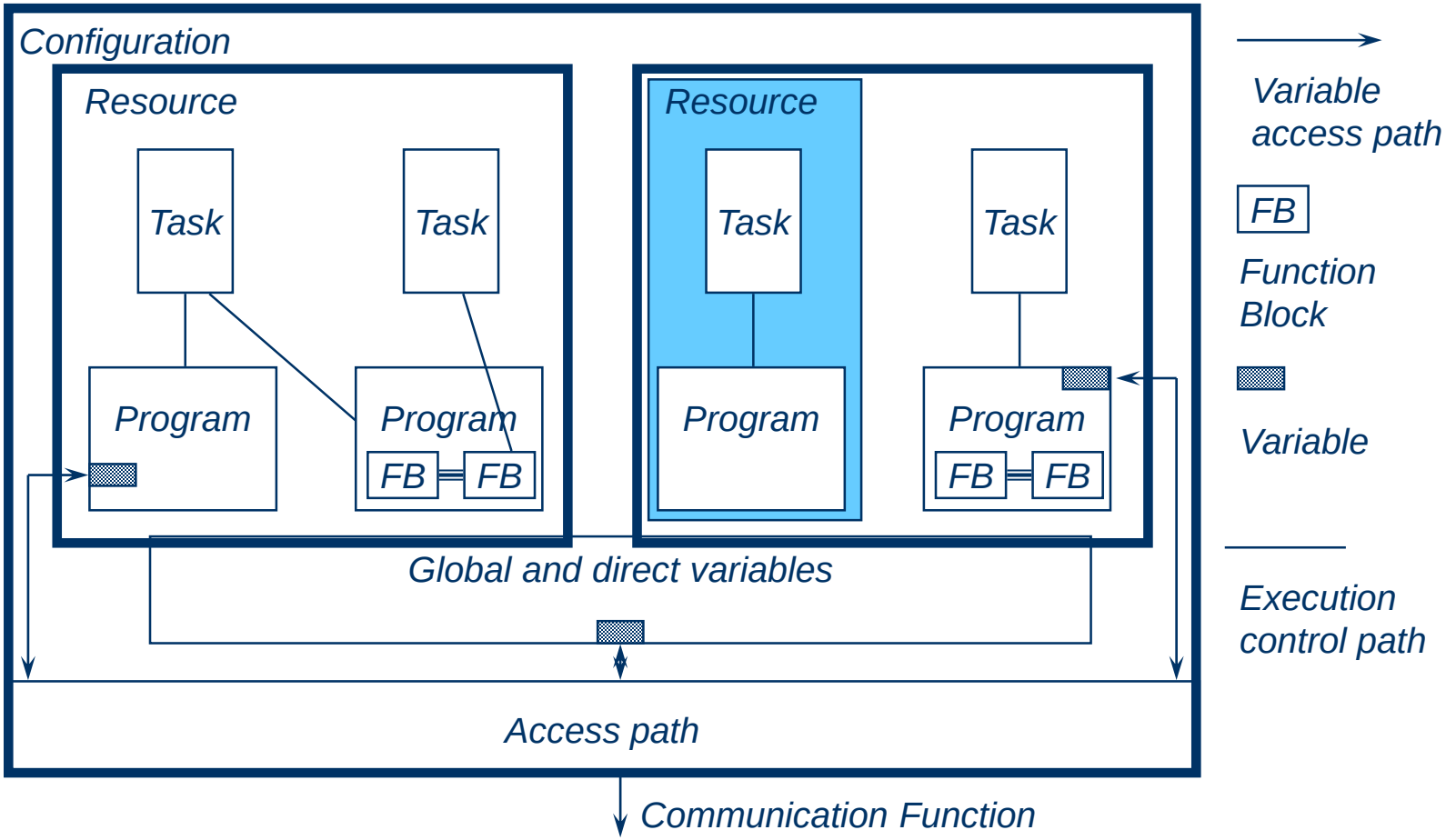
- O escopo da variável é local ao elemento de software que as declara, permitindo acesso somente ao elemento.
- O escopo da variável é global sendo acessada por todos os elementos contidos no mesmo nível.

Caminhos de acesso:

- Permitem a transferência de dados entre diferentes configurações.

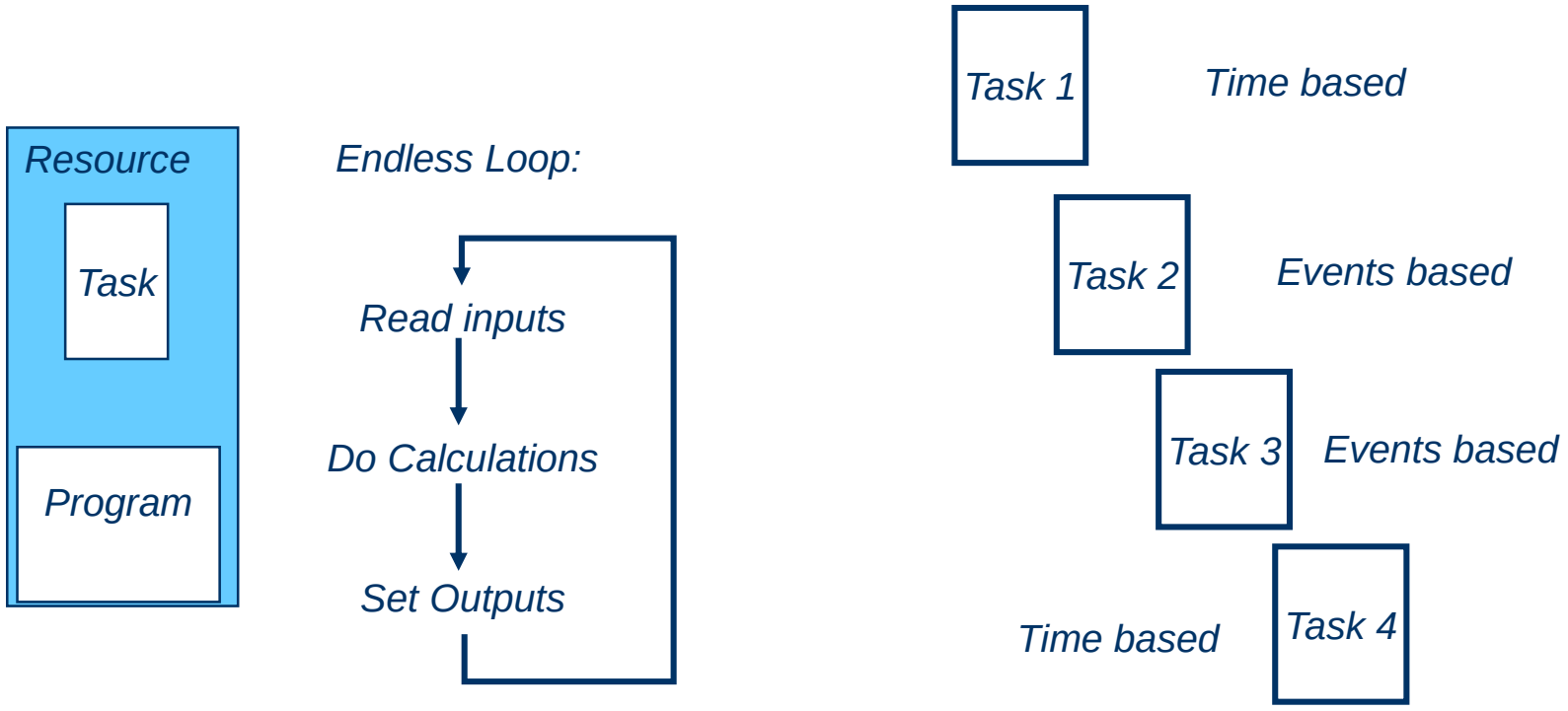
IEC 61131-3 : Elementos Comuns
Modelo de Software

Vantagens do Modelo IEC 61131-3



IEC 61131-3 : Elementos Comuns
Modelo de Software

CP Conventional vs IEC 61131-3



IEC 61131-3 : Elementos Comuns

ELEMENTOS COMUNS

- Configurações
- Recursos
- Tarefas
- Programming Organization Units(POUs)
  - * Functions
  - * Function Blocks
  - * Programs

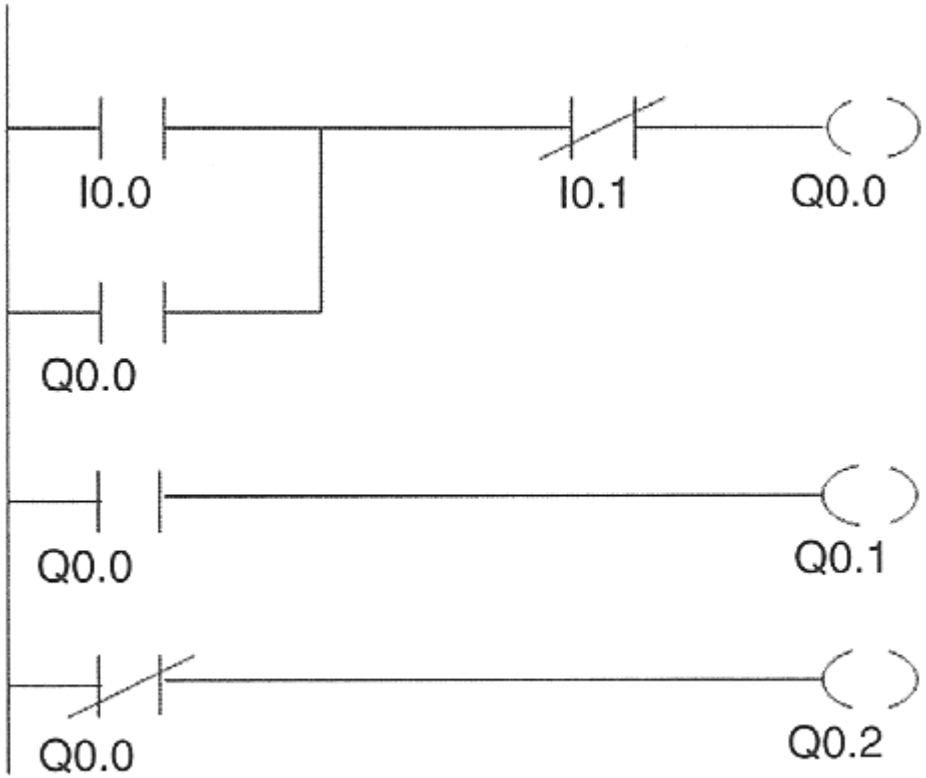
Unidade de Organização de Programa (POU)

Programa

- Pode ser construído a partir de diferentes elementos de software escritos em qualquer uma das diferentes linguagens 61131.
- Um programa consiste de um código de execução, capaz de trocar dados através das conexões de software.
- Um programa pode acessar as variáveis do CLP e se comunicar com outros programas.
- A execução de diferentes partes de um programa pode ser controlada usando *Tasks*.

Unidade de Organização de Programa (POU)

- Exemplo programa linguagem Ladder (LD)



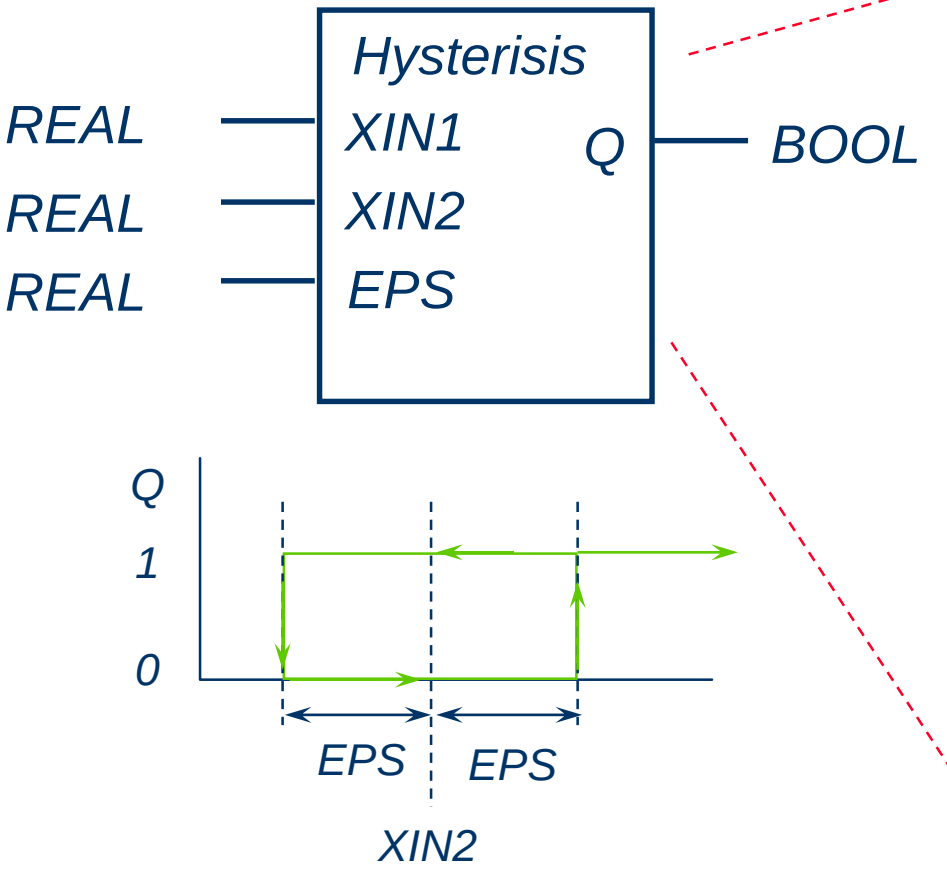
Unidade de Organização de Programa (POU)

Bloco Funcional (FB)

- O conceito de FBs é um dos mais importantes da norma IEC61131-3, para permitir o projeto de software de forma hierárquica e estruturada.
- FB podem ser utilizados para a criação de elementos de software totalmente reutilizáveis.
- FB possuem um conjunto de dados, os quais podem ser alterados por um algoritmo interno.
- O conjunto de dados é mantido na memória para uma determinada instância do bloco funcional.

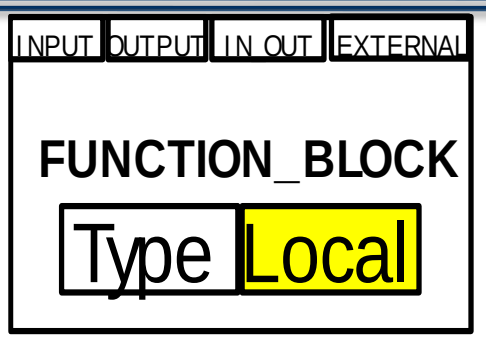
Unidade de Organização de Programa (POU)

- Function Block example



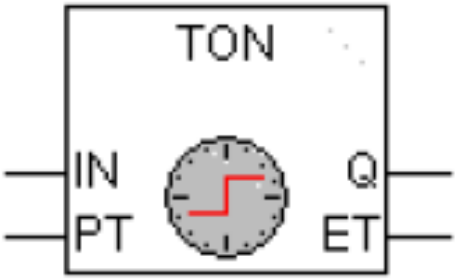

```
FUNCTION_BLOCK HYSTERISIS
 VAR_INPUT
   XIN1, XIN2 : REAL;
   EPS : REAL;  (* Hysterisis band *)
 END_VAR
 VAR_OUTPUT
   Q : BOOL := 0
 END_VAR
 IF Q THEN
   IF XIN1 < (XIN2-EPS) THEN
     Q := 0 (* XIN1 decreasing *)
   END_IF;
 ELSIF XIN1 > (XIN2 + EPS ) THEN
   Q := 1; (* XIN1 increasing *)
 END_IF;
END_FUNCTION_BLOCK
```

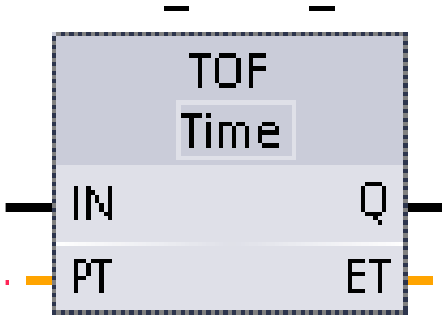
Unidade de Organização de Programa (POU)

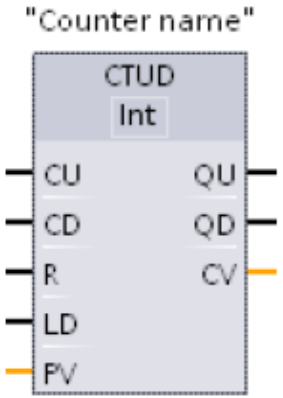


Function Blocks:

- Standard Function Blocks



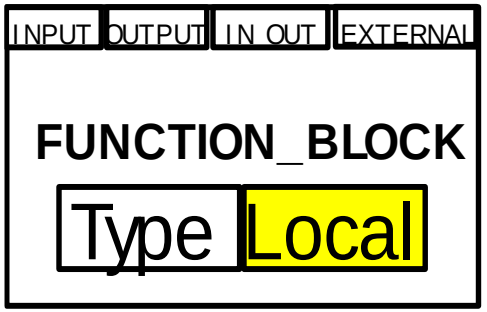


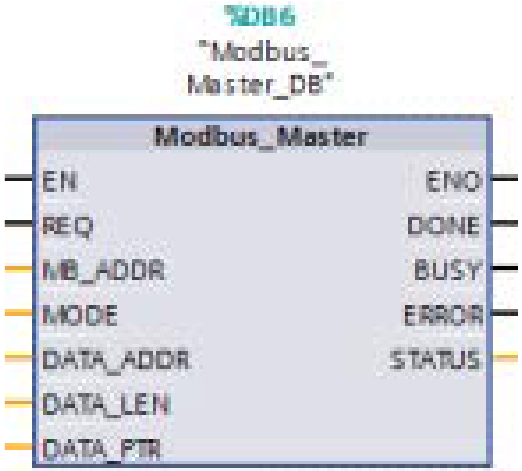


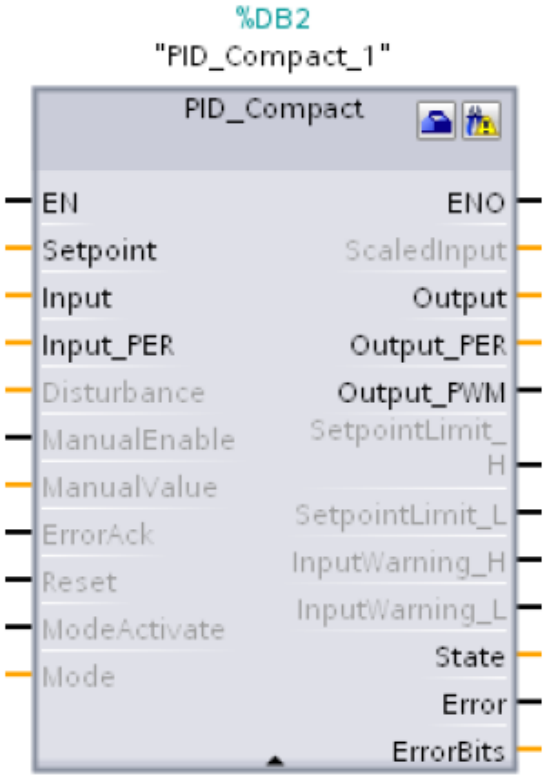
Unidade de Organização de Programa (POU)

Function Blocks:

- Standard Function Blocks
- FB adicionais (fabricante)



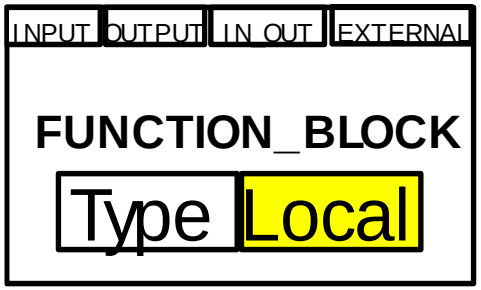


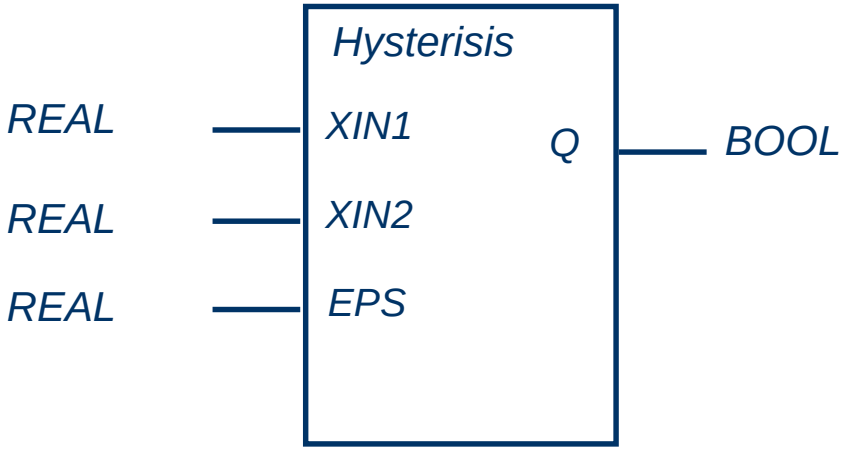


Unidade de Organização de Programa (POU)

Function Blocks:

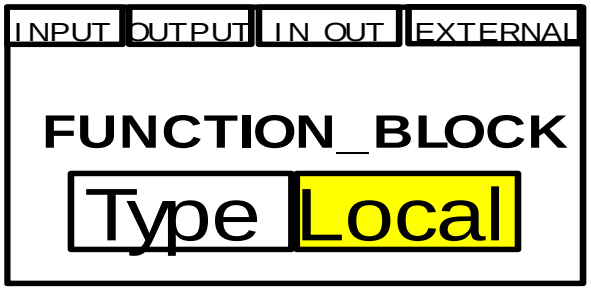
- Standard Function Blocks
- FB adicionais (fabricante)
- FB definidos pelo usuário

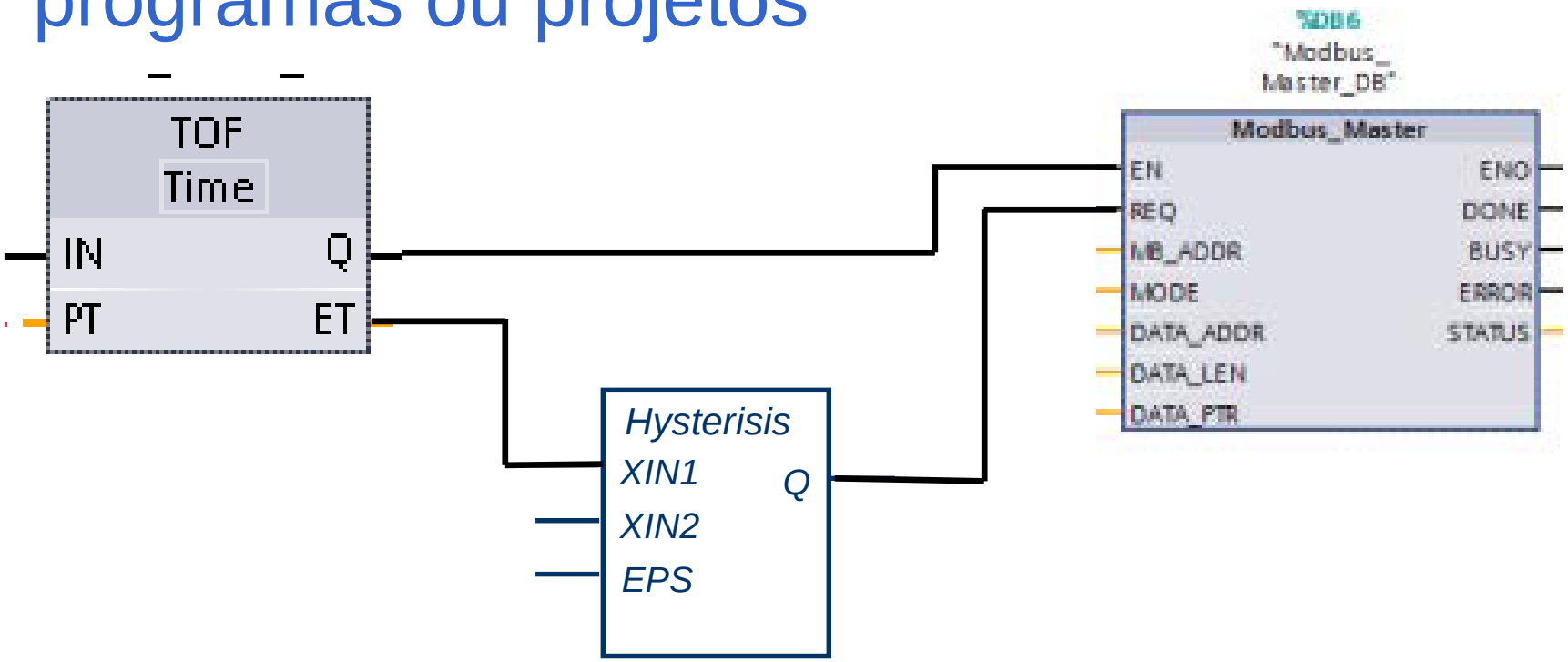




Unidade de Organização de Programa (POU)

- Todos FB são altamente reutilizáveis no mesmo programa ou diferentes programas ou projetos





Unidade de Organização de Programa (POU)

Funções:

- São elementos de software que desempenham um determinado algoritmo e tem característica dinâmica (não possuem persistência).
- São os elementos mais básicos do software, como exemplos blocos aritméticos, lógicos, comparadores.

Unidade de Organização de Programa (POU)

* Funções Padrões

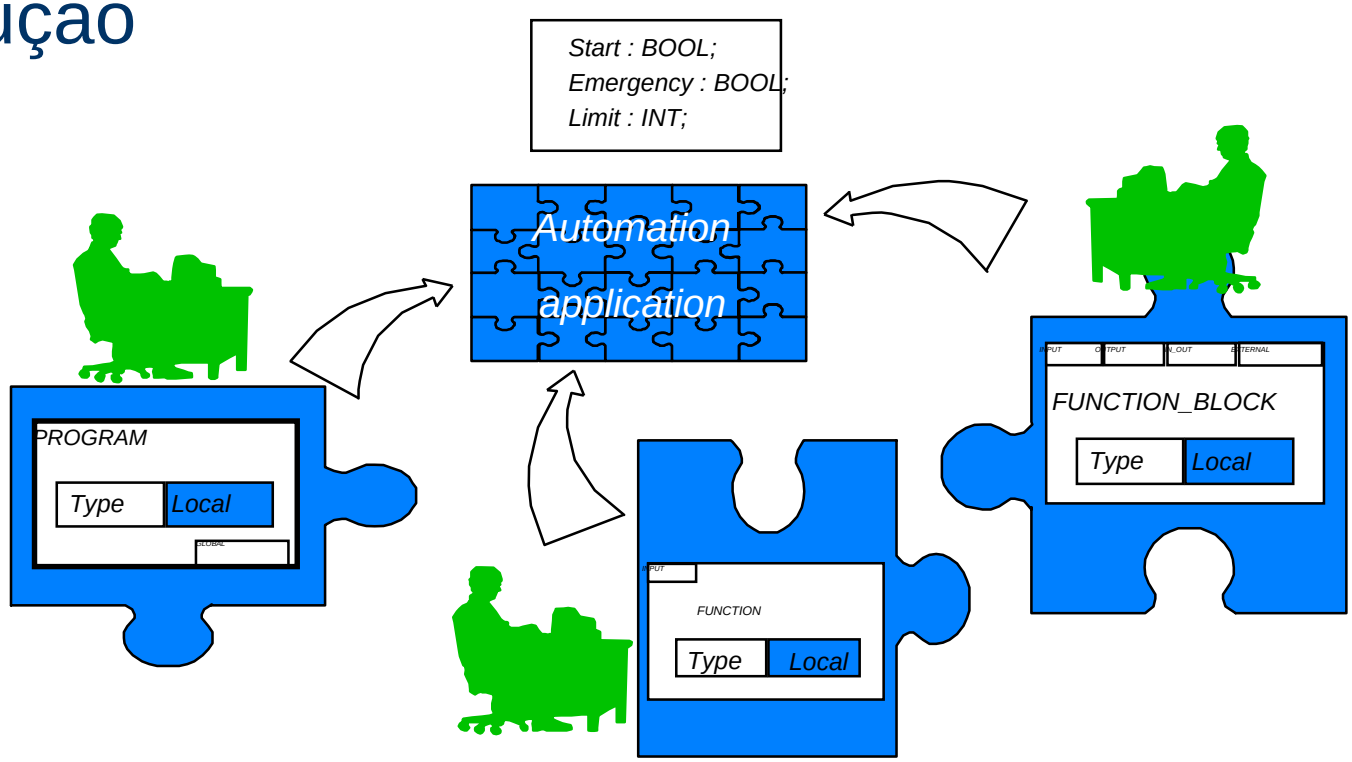
ADD, SQRT, SIN, COS, GT, MIN, MAX, AND, OR, etc.

- Funções do Usuário

```
FUNCTION SIMPLE_FUN : REAL
        VAR_INPUT
                A, B      : REAL;
                C                   : REAL := 1.0;
        END_VAR
        SIMPLE_FUN := A*B/C;
END FUNCTION
```

Unidade de Organização de Programa (POU)

POUs : Projeto de integração como tijolos de construção



Unidade de Organização de Programa (POU)

Vantagens das POUs:

- Criar bibliotecas próprias de FB (por área de aplicação)
- FBs são modularizados, testados e documentados
- Tornar as bibliotecas acessíveis e reutilizável em qualquer lugar
- Alterar a programação para criar redes de FBs
- Economize 40% no próximo projeto

IEC 61131-3 Standard

Elementos Comuns

Linguagens de Programação

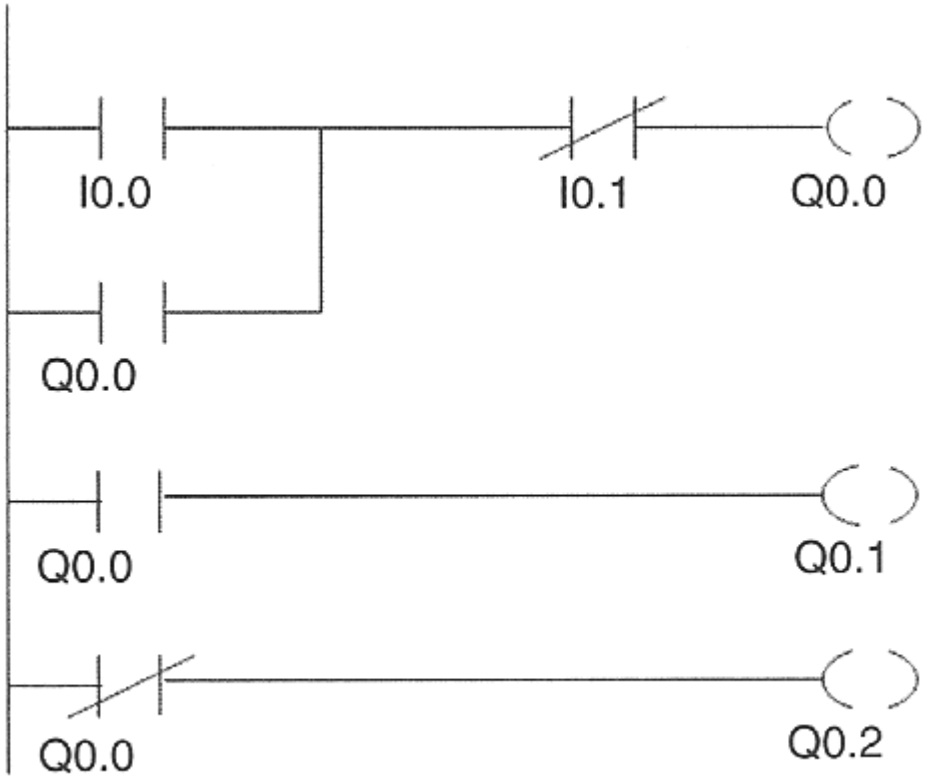
Linguagens de Programação

- A especificação IEC 61131-3 define cinco linguagens de programação:

Texto Estruturado (ST)	Textual
Lista de Instruções (IL)	Textual
Diagrama de blocos Funcionais (FBD)	Gráfica
Diagrama Ladder (LD)	Gráfica
Sequenciamento Gráfico de Funções (SFC)	Gráfica

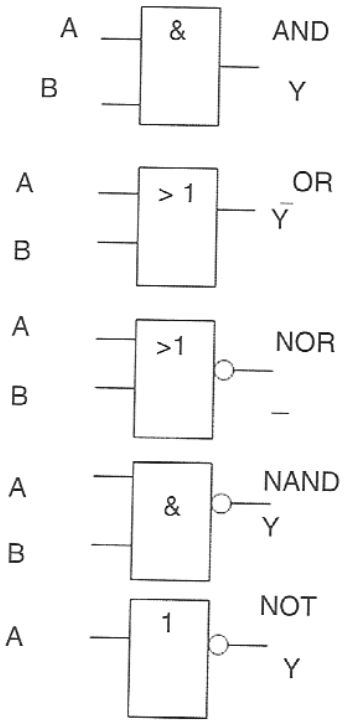
Linguagens de Programação

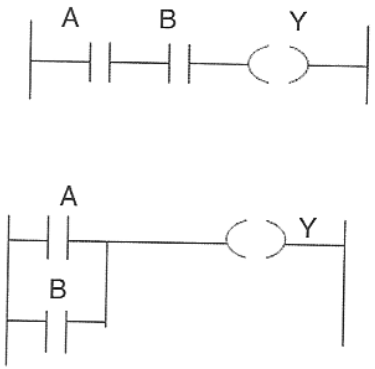
- Exemplo programa linguagem Ladder (LD)



Linguagens de Programação

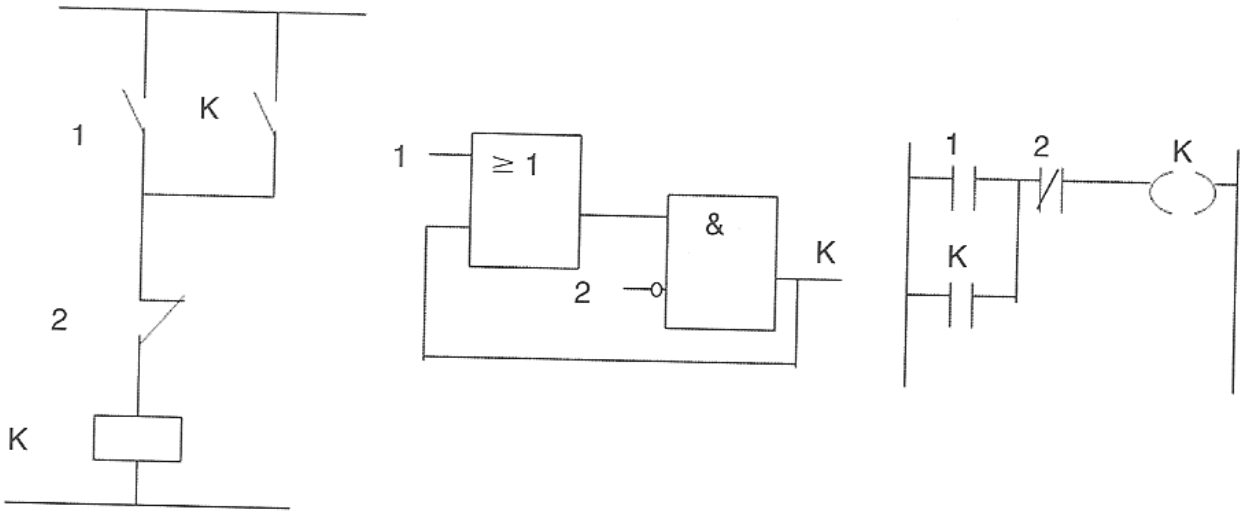
- Exemplo programa linguagem Function Block (FBD)





Linguagens de Programação

- Exemplo programa linguagem Function Block (FBD)



Linguagens de Programação

Linguagem Instruction List (IL)

- Poderosa linguagem de programação
- Difícil entendimento
- Baseado na linguagem Assembler

```
NETWORK  1 // Partida motor
LD  I0.1              // Se a entrada I0.1 é ativa (on)
A    I0.2              // E se a entrada I0.2 é ativa (on)
= Q0.2               // Partida motor 2
NETWORK  2 // Parada de emergência
LD I0.3            //Se a entrada I0.3 é ativa (on)
ON I0.4          // Ou então a entrada I0.4 não é ativa (off)
R  Q 0.1, 1    //  Para motor 1
```

Instrução — Operando — Comentário inicia com duas barras oblíquas

Linguagens de Programação

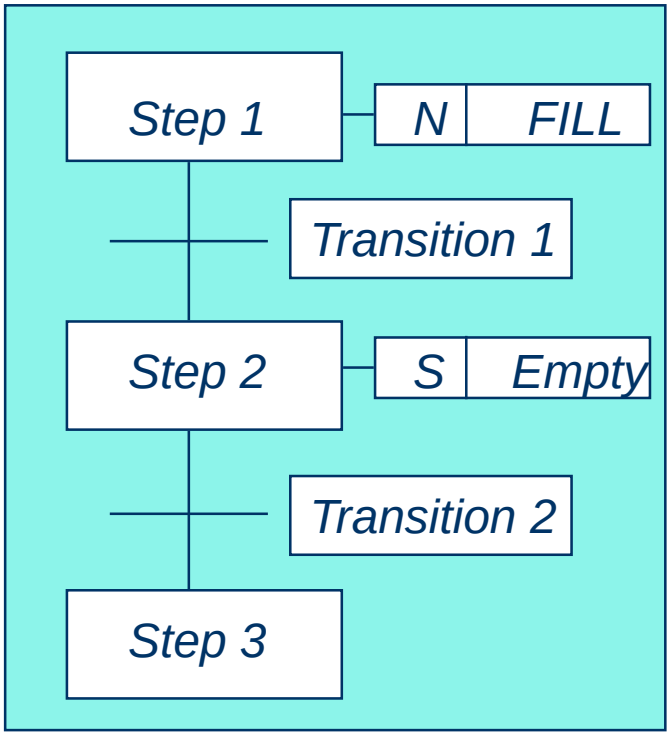
Linguagem Struct Text (ST)

- Poderosa linguagem de programação estruturada
- Baseada na linguagem Pascal
- Permite operações não existentes em LD e FBD

```
10  IF  I0.0=I0.1   //  Se a entrada I0.0 é igual a entrada I0.1
20  THEN  SET Q0.0  // Ativa (on) a saída Q0.0
30   ELSE RES Q0.0   //  Se não desativa (off) a saída Q0.0
40  GOTO  10   //  Salta a instrução 10
```

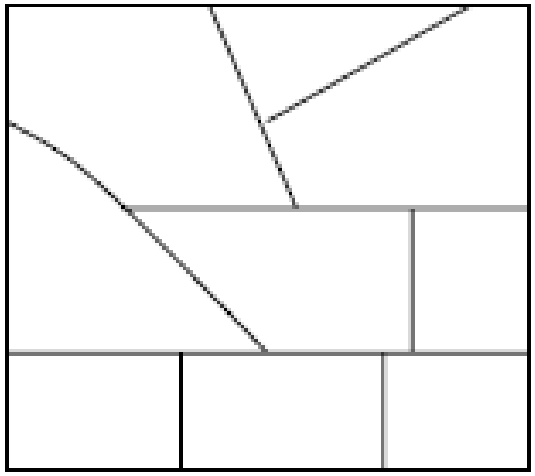
Linguagens de Programação

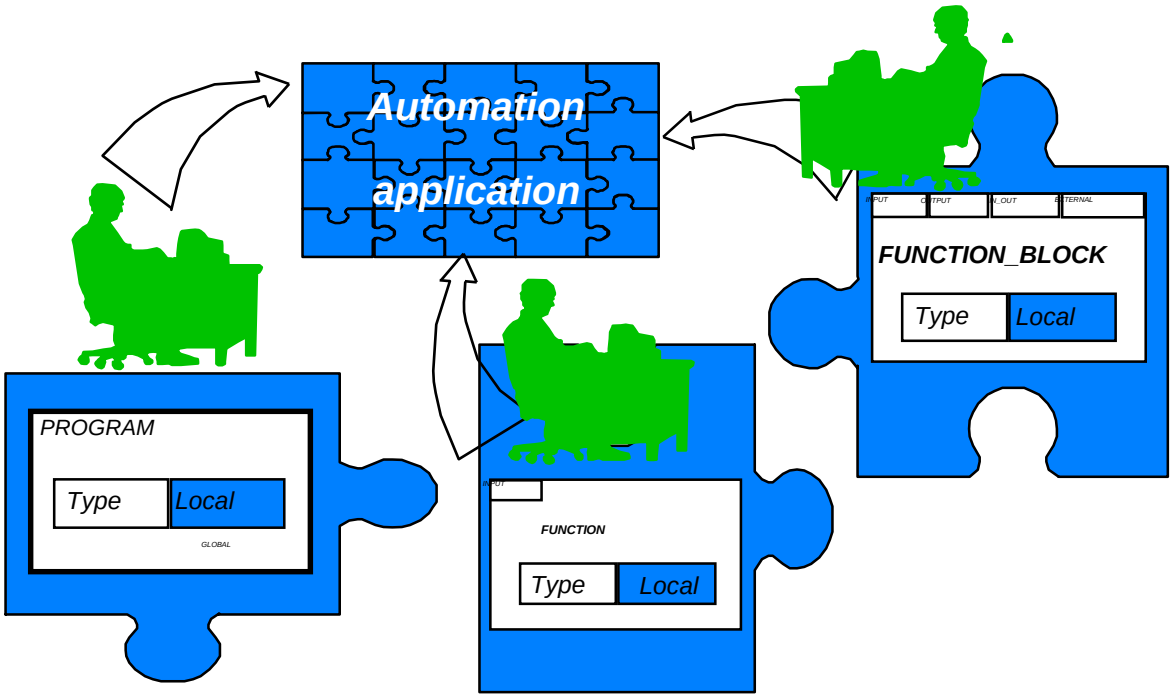
- Exemplo programa linguagem Sequencial Function Chart (SFC)



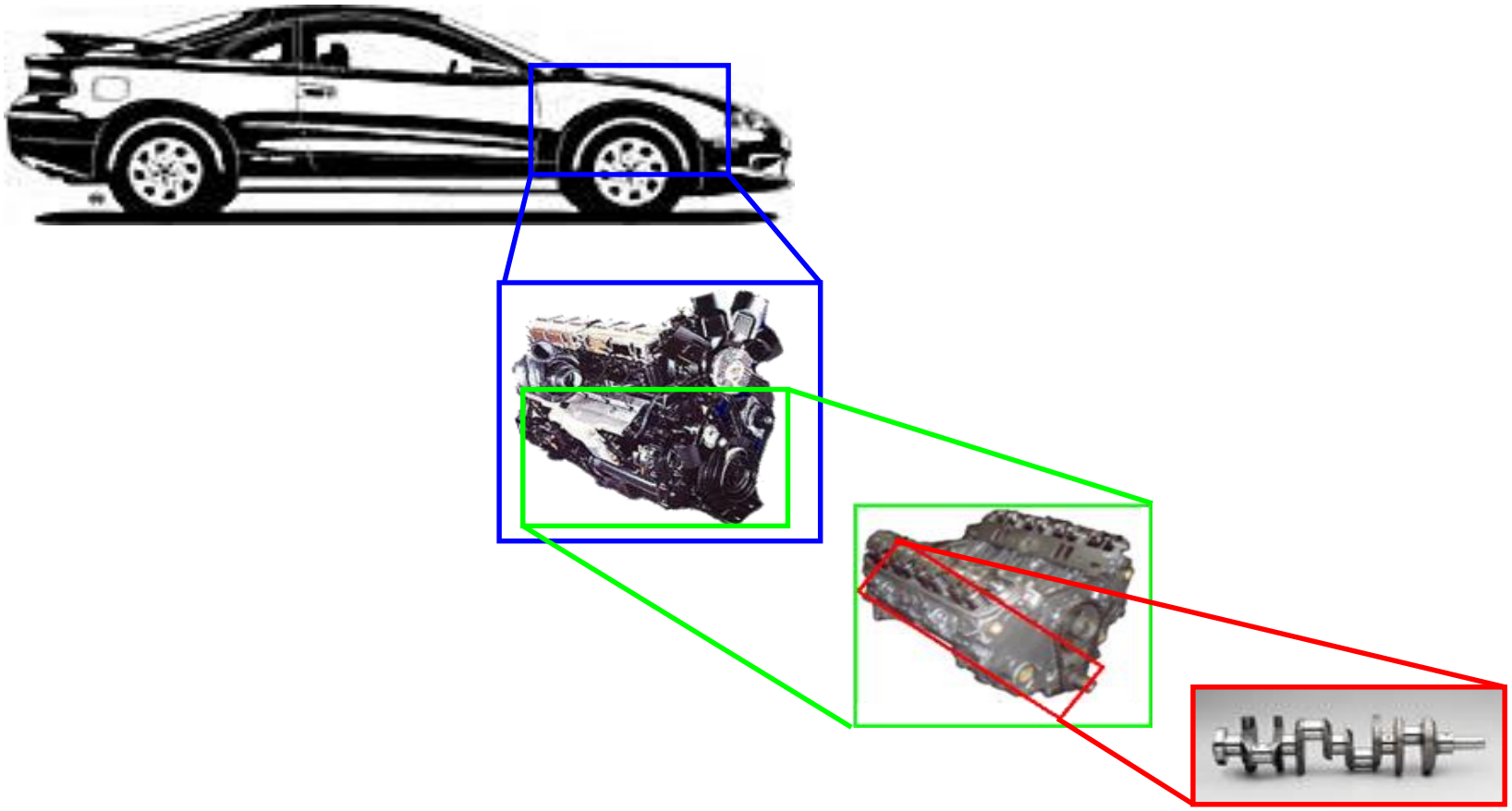
- Técnica gráfica poderosa para descrever o comportamento seqüencial de um programa
- Utilizado para particionar um problema de controle
- Mostra a visão geral, também adequada para diagnósticos rápidos

Decomposição e reuso



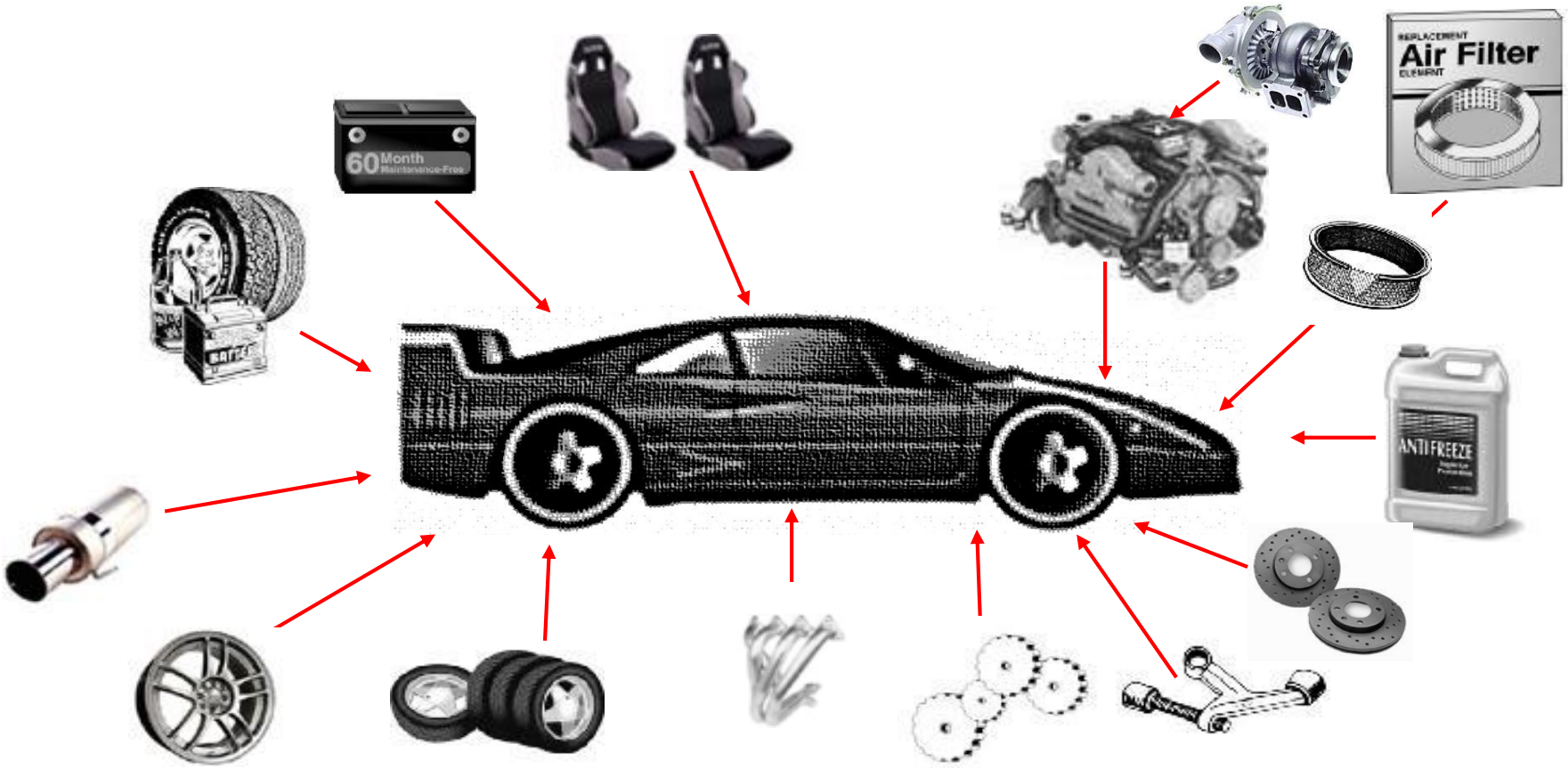


Decomposição



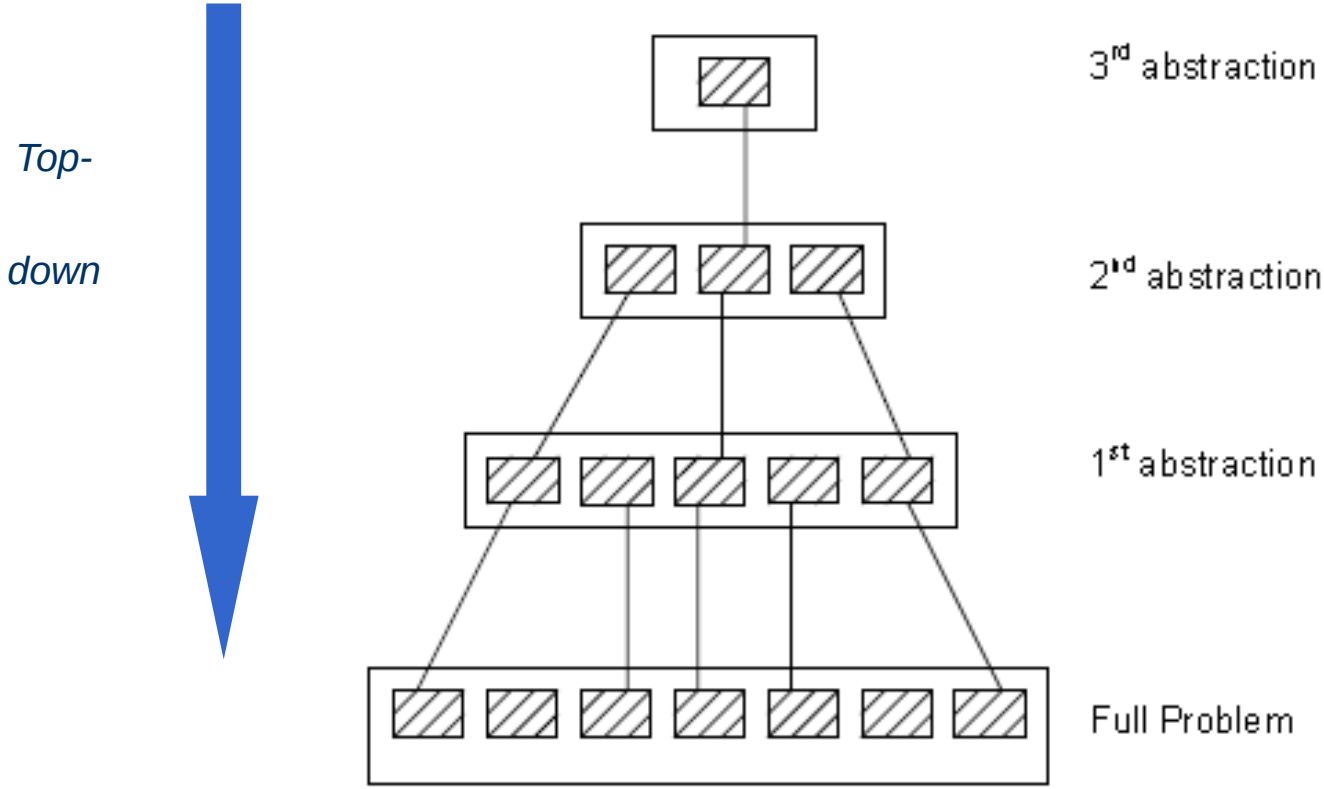
Reuso via Function Blocks padrões

Pode-se montar um carro através dos diferentes componentes padronizados.



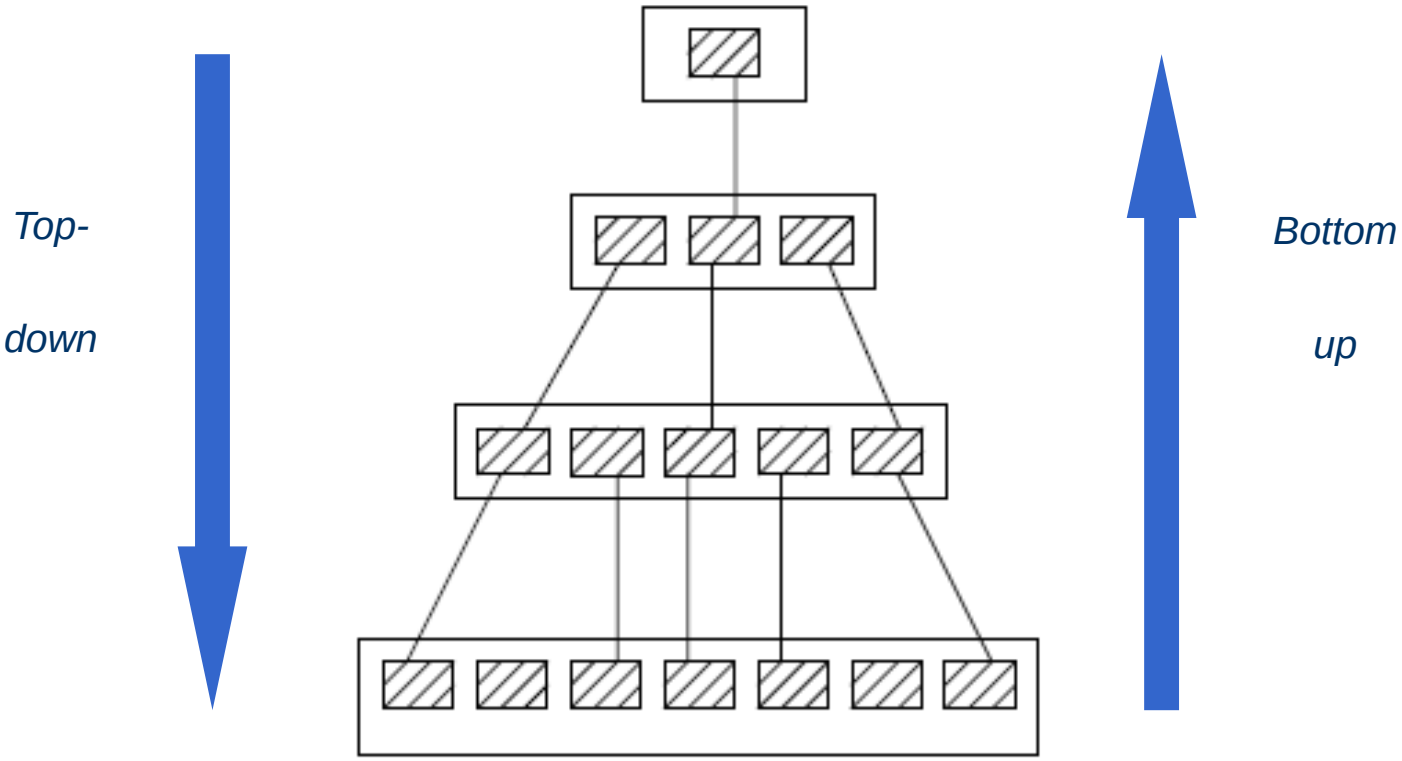
Abstração e Hierarquia

A decomposição pode fornecer diferentes níveis de abstração, que fornecem maneiras diferentes de analisar o "problema".



Bottom-up após top-down

- Primeiramente decompõe-se o problema. Então preenche cada uma das caixinhas.

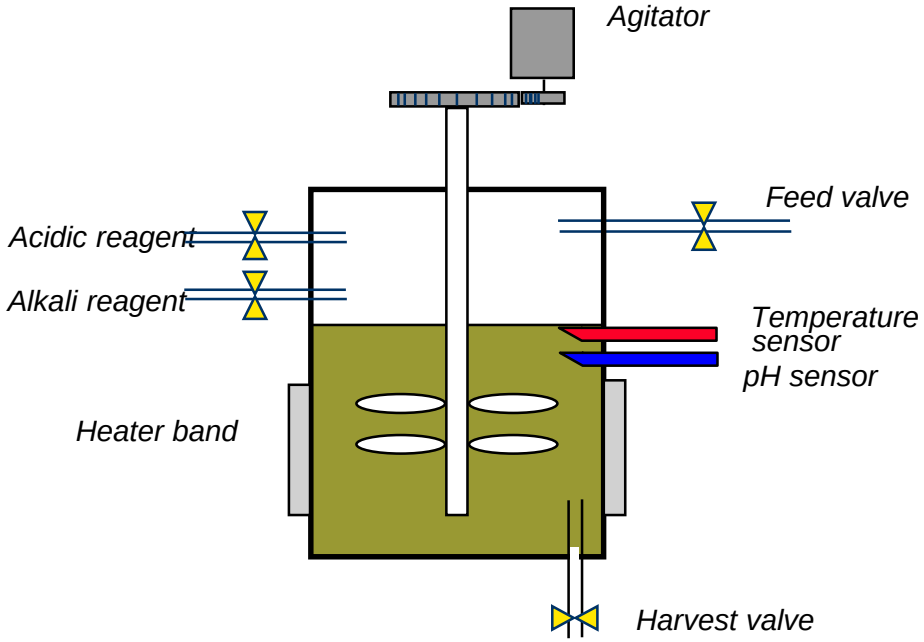


Como fazer isso no IEC 61131-3 ?

ESTUDO DE CASO DE UM SISTEMA DE FERMENTAÇÃO

Processo de fermentação

- Um tanque é alimentado pela válvula de alimentação com o líquido. Pode ser aquecido através do sistema de aquecimento. Pode ser agitado através do motor do reator,e possui entradas para fluido ácido e alcalino.

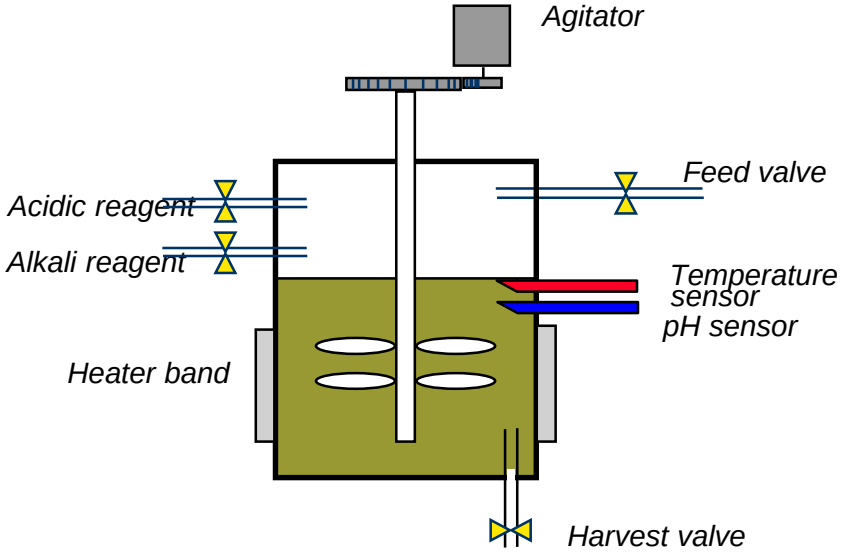


Como criar um sistema de controle para este sistema?

Etapa 1: Identificação de Interfaces Externas no Sistema

Identificação das entradas:

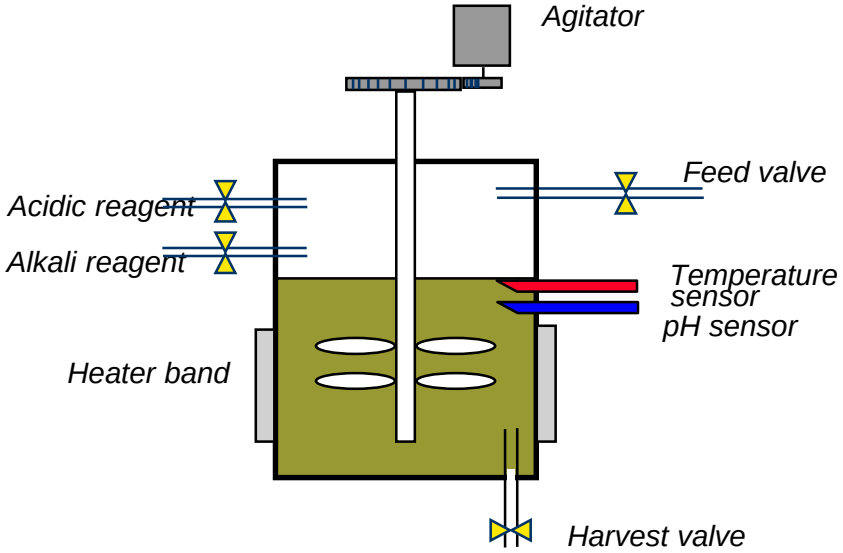
- Sensor de temperatura
- Sensor de pH
- Posições das válvulas
- Velocidade do motor



Etapa 1: Identificação de Interfaces Externas no Sistema

Identificação das saidas:

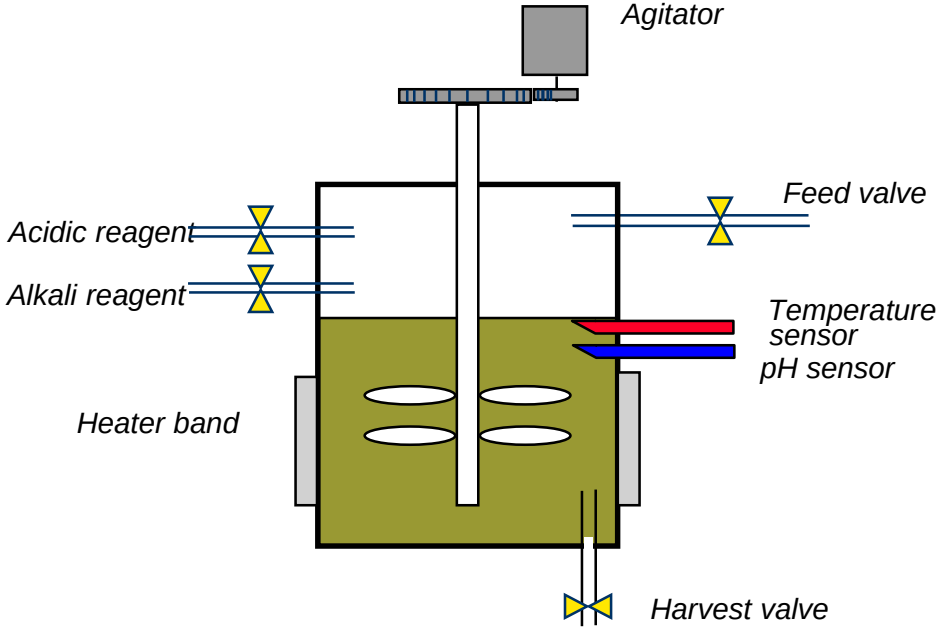
- Posição das válvulas
- Posição do Motor
- Banda de aquecimento



Etapa 2: Definição dos principais sinais entre o sistema e a planta

- Neste exemplo não há acoplamento à planta,

mas poderia haver algo como:
... acoplando-se aos vasos principais com líquidos
… acoplando ao sistema de transporte / estação de enchimento após o fim do processo



Etapa 3: Definição de todas as interações do operador, substituições e dados de supervisão

Para o operador, definimos:

… Um botão "Iniciar"

… Um botão "Parar"

… Uma entrada de "duração"

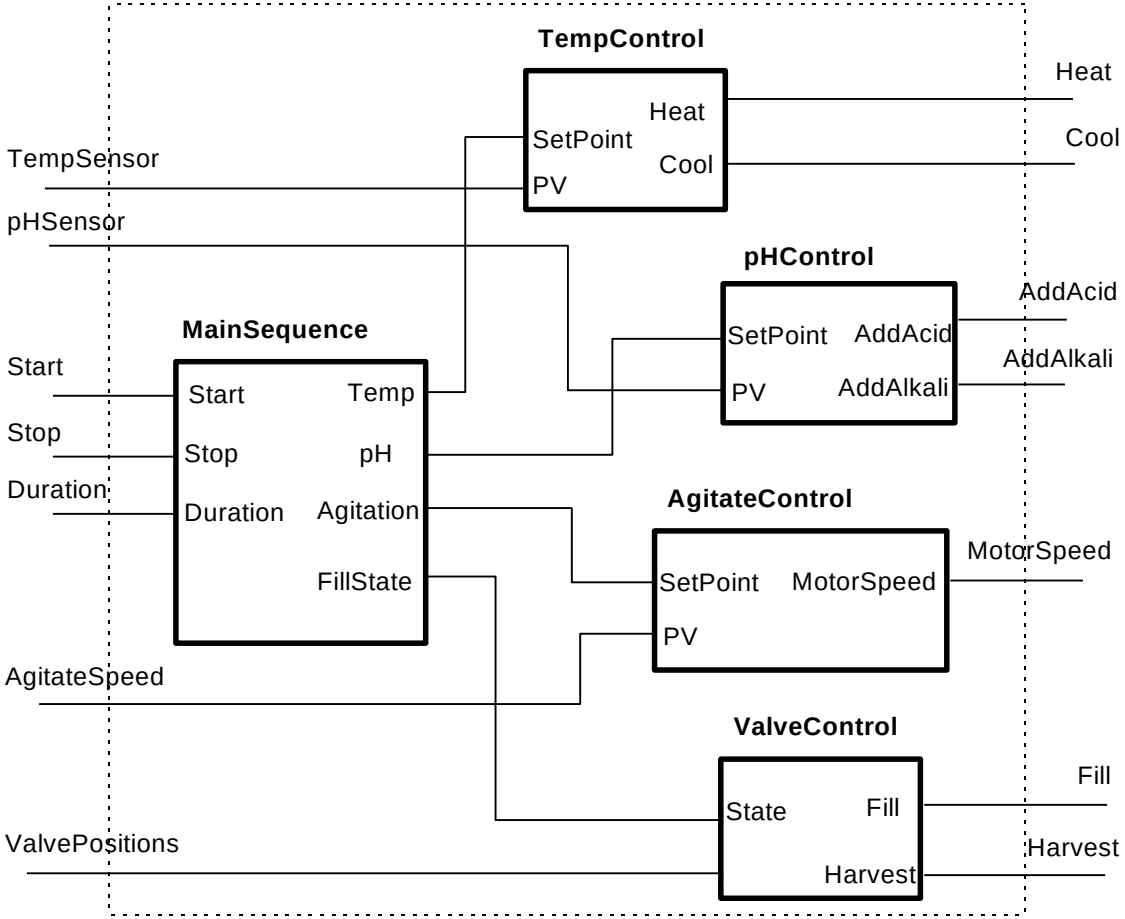
Agora nós definimos todas as interfaces

Etapa 4: Quebra em partições lógicas

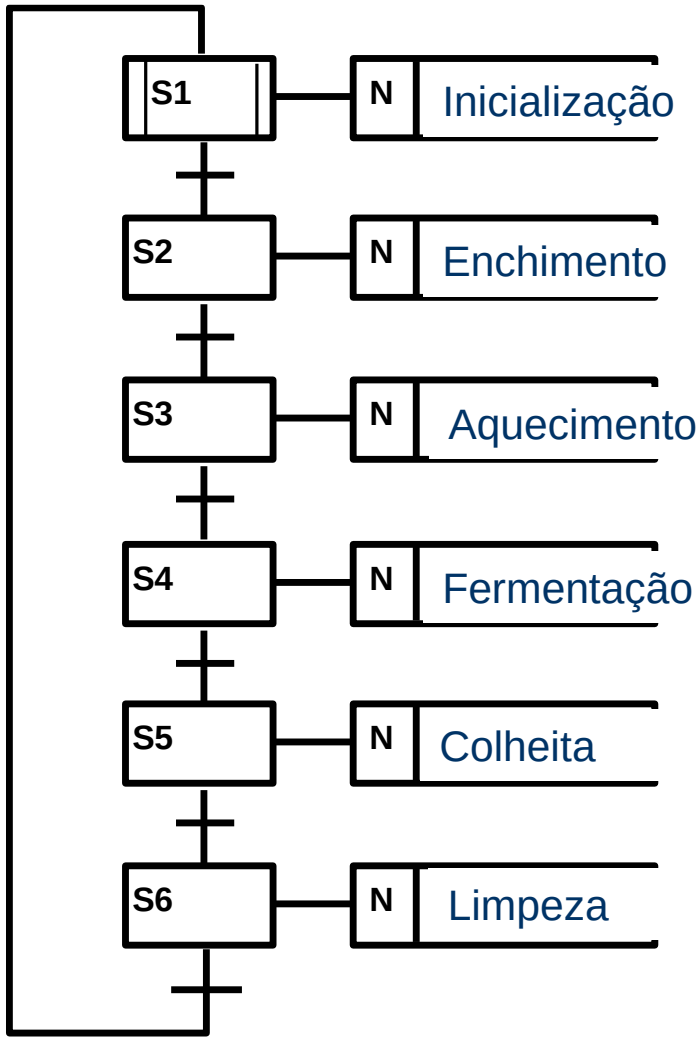
- Sequencia principal - enchimento, aquecimento, agitação, fermentação, colheita, limpeza.
- Controle de válvulas
- Controle de temperatura
- Controle do Agitador
- Controle do PH

Passo 5 : Definição dos programas (POU)

- A estrutura do programa de controle de fermentação seria:



Sequencia principal usando SFC



- Onde as Actions e Transitions pode ser programada em qualquer uma das quarto linguagens IEC